

## **IN THE CLAIMS**

Claim 1 (canceled)

Claim 2 (currently amended): A method of launching a software application in a hand-held device, comprising:

- receiving an abbreviated textual command in a natural language search engine, entered by a user of the hand-held device; and

- while receiving the abbreviated textual command ~~and without the user having entered a delimiter denoting an end to entry of the abbreviated textual command~~, performing the steps of:

- searching a natural language database that stores a data set of abbreviated textual commands and associated application commands;

- analyzing historical preferences to determine one or more probable complete commands matching a currently received portion of the abbreviated textual command; and

- displaying a list of probable complete commands matching the currently received portion of the abbreviated textual command.

Claim 3 (previously presented): The method of claim 2, comprising the additional step of:

- if a user selects a complete command from the list, then setting the complete command as the abbreviated textual command, and executing the associated application command.

Claim 4 (previously presented): The method of claim 2, comprising the additional step of:

- if a user does not select a complete command from the list, then receiving an entire abbreviated textual command in the natural language search engine.

Claim 5 (previously presented): The method of claim 4, further comprising:

- if the abbreviated textual command has an exact match in the data set, then setting the exact match as a user command;

- if the abbreviated textual command does not have an exact match in the data set, then analyzing historical preferences to determine if the abbreviated textual command has a probable match in the data set;

- if the abbreviated textual command has a probable match in the data set, then setting the probable match as the user command;

if the abbreviated textual command does not have a probable match in the data set, then presenting a list of possible commands, receiving a command choice, and setting the command choice as the user command; and  
executing the user command.

Claim 6 (previously presented): The method of claim 2, wherein the step of analyzing historical preferences is performed using a set of probability factors that are generated based on historical preferences, where the abbreviated textual command has a probable match in the data set when a probability factor associated with the probable match is greater than a predetermined value.

Claim 7 (previously presented): The method of claim 6, wherein the predetermined value is defined by a user.

Claim 8 (previously presented): The method of claim 6, comprising the additional step of:  
adjusting the set of probability factors each time the abbreviated textual command is entered into the hand-held device.

Claim 9 (previously presented): The method of claim 2, wherein:  
the abbreviated textual command has a first component and a second component, wherein the first component represents a desired application command, and the second component represents a desired application tag; and  
the natural language database stores a data set of abbreviated textual commands and associated application commands and tags.

Claim 10 (previously presented): The method of claim 2, wherein the abbreviated textual command is entered into a graphical dialog box.

Claim 11 (previously presented): The method of claim 2, wherein the natural language search engine can receive the abbreviated textual command while any of the software applications are executing.

Claim 12 (previously presented): The method of claim 5, wherein the list of possible commands presented if the abbreviated textual command does not have a probable match in the data set includes a set of recently executed application commands.

Claim 13 (previously presented): The method of claim 5, wherein the list of possible commands presented if the abbreviated textual command does not have a probable match in the data set includes a set of generic application commands that the natural language search engine is capable of executing.

Claims 14-36 (canceled)

Claim 37 (currently amended): A method comprising:

- storing a data set of abbreviated textual commands and corresponding complete commands;
- receiving a portion of an abbreviated textual command being entered by a user; and
- before receiving the entire abbreviated textual command, ~~and without the user having entered a delimiter denoting an end to entry of the abbreviated textual command,~~ comparing the received portion of the abbreviated textual command to the stored abbreviated commands to determine a probable subset of the complete commands.

Claim 38 (previously presented): The method of claim 37 further comprising after the comparing step:

- displaying the probable subset of the complete commands to the user.

Claim 39 (previously presented): The method of claim 38 further comprising after the displaying step:

- receiving an indication of which of the displayed complete commands a user chooses; and
- executing the chosen complete command.

Claim 40 (previously presented): The method of claim 38 further comprising after the displaying step:

- receiving a further portion of the abbreviated textual command; and
- narrowing the probable subset based on the further portion received.

Claim 41 (previously presented): The method of claim 37 further comprising:

when the probable subset consists of only one complete command, executing that one complete command.

Claim 42 (previously presented): The method of claim 37 wherein the storing step includes a user assigning which complete commands should correspond in the future to which abbreviated textual commands.

Claim 43 (previously presented): The method of claim 37 wherein the storing step includes generating the data set based on which abbreviated textual commands a user has historically used for choosing each complete command.

Claim 44 (previously presented): The method of claim 37 wherein the comparing step includes:

if the data set indicates that the user has chosen to execute a particular complete command more than a predetermined percentage of the time less than 100% after having entered an abbreviated textual command matching the currently received portion of text, then narrowing the subset to that command.

Claim 45 (previously presented): The method of claim 44 wherein the predetermined percentage is 50%.

Claims 46-47 (canceled)

Claim 48 (currently amended): A method comprising:

receiving a text string being entered by a user; ~~and~~  
while receiving the text string, comparing a received portion of the text string to stored text  
commands to determine which ~~one~~ of the stored text commands is a probable text command based  
on a portion of the probable text command matching the received text string; and  
initiating a software operation corresponding to the probable text command;  
~~the comparing and initiating steps being performed without the user having entered a~~  
~~delimiter denoting an end of the text string.~~

Claim 49 (previously presented): The method of claim 48 wherein said portion of the probable text command is not the entire text command.

Claim 50 (previously presented): The method of claim 48 wherein the comparing step includes:  
    identifying a plurality of the stored text commands that have portions matching the received text string; and  
    determining which one of the plurality is the probable text command based on historical preferences.